IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **DeWitt, Jr. et al.**	§	
	§	Group Art Unit: **2185**
Serial No. **10/757,212**	§	
	§	Examiner: **Savla, Arpan P.**
Filed: **January 14, 2004**	§	
	§	
For: **Method and Apparatus for Generating Interrupts Based on Arithmetic Combinations of Performance Counter Values**	§	

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

35525
PATENT TRADEMARK OFFICE
CUSTOMER NUMBER

APPEAL BRIEF (37 C.F.R. 41.37)

This brief is in furtherance of the Notice of Appeal, filed in this case on October 10, 2006.

A fee of $500.00 is required for filing an Appeal Brief. Please charge this fee to IBM Corporation Deposit Account No. 09-0447. No additional fees are believed to be necessary. If, however, any additional fees are required, I authorize the Commissioner to charge these fees which may be required to IBM Corporation Deposit Account No. 09-0447. No extension of time is believed to be necessary. If, however, an extension of time is required, the extension is requested, and I authorize the Commissioner to charge any fees for this extension to IBM Corporation Deposit Account No. 09-0447.

REAL PARTY IN INTEREST

The real party in interest in this appeal is the following party: International Business Machines Corporation of Armonk, New York.

RELATED APPEALS AND INTERFERENCES

With respect to other appeals or interferences that will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal, there are no such appeals or interferences.

STATUS OF CLAIMS

A. TOTAL NUMBER OF CLAIMS IN APPLICATION

Claims in the application are: 1-23.

B. STATUS OF ALL THE CLAIMS IN APPLICATION

1. Claims canceled: None.
2. Claims withdrawn from consideration but not canceled: None.
3. Claims pending: 1-23.
4. Claims allowed: None.
5. Claims rejected: 1-23.
6. Claims objected to: None.

C. CLAIMS ON APPEAL

The claims on appeal are: 1-23.

STATUS OF AMENDMENTS

There are no amendments after the final rejection.

SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 1:

The present invention provides a method in a data processing system for processing instructions. (Specification, page 81, lines 4-13) The present invention receives a threshold value and an identification of a plurality of addresses to be monitored during the execution of a computer program. (Specification, page 86, line 27 to page 87, line 1) The present invention associates hardware counters with the plurality of addresses. (Specification, page 87, lines 4-6) The present invention executes the computer program and incrementing respective hardware counters when the plurality of addresses are accessed and a performance indicator associated with the plurality of addresses is encountered. (Specification, page 87, lines 4-6) The present invention performs an action in response to a determination that a predefined relationship between the threshold value and a combination of values obtained from the hardware counters is present. (Specification, page 87, lines 11-25)

Dependent claim 2:

The present invention arithmetically combines values of the hardware counters to generate a combined counter value. (Specification, page 87, lines 14-15) The present invention compares the combined counter value to the threshold value. (Specification, page 87, lines 15-16) The present invention performs the action in response to a relationship between the combined counter value and the threshold value being present. (Specification, page 87, lines 20-23)

Dependent claim 3:

The present invention generates an interrupt if the predetermined relationship between the combined counter value and the threshold value is present. (Specification, page 87, lines 20-23)

Dependent claim 4:

The present invention provides for the steps of arithmetically combining values of the hardware counters, comparing the combined counter value, and performing the action are performed in response to incrementing a hardware counter. (Specification, page 87, lines 4-25; page 81, lines 4-13; and page 81, line 23 to page 82, line 19)

Dependent claim 5:

The present invention provides for the steps of arithmetically combining values of the hardware counters, comparing the combined counter value, and performing the action are performed within microcode of a processor of the data processing system. (Specification, page 87, lines 4-25; page 81, lines 4-13; and page 81, line 23 to page 82, line 19)

Dependent claim 6:

The present invention sends the interrupt to an interrupt handler of a performance monitoring application, wherein the interrupt handler performs an operation based on receipt of the interrupt. (Specification, page 87, lines 23-24; page 81, lines 4-13; and page 84, lines 6-10)

Dependent claim 7:

The present invention provides for the operation is at least one of generating a log entry in a performance monitoring application log and notifying a log daemon process of an event. (Specification, page 84, lines 9-14)

Dependent claim 8:

The present invention provides for arithmetically combining values of the hardware counters includes combining values in accordance with a condition indicated by a performance monitoring application. (Specification, page 81, lines 4-13; and page 84, lines 19-24)

Independent claim 9:

The present invention provides for a computer program product in a computer readable recordable-type medium for processing instructions. (Specification, page 81, lines 4-13) The present invention provides first instructions for receiving a threshold value and an identification of a plurality of addresses to be monitored during the execution of a computer program. (Specification, page 86, lines 27 to page 87, line 1) The present invention provides second instructions for associating hardware counters with the plurality of addresses. (Specification, page 87, lines 4-6) The present invention provides third instructions for executing the computer program and incrementing respective hardware counters when the plurality of addresses are accessed and a performance indicator associated with the plurality of addresses is encountered. (Specification, page 87, lines 4-6) The present invention provides fourth instructions for performing an action in response to a determination that a predefined relationship between the threshold value and a combination of values obtained from the hardware counters is present. (Specification, page 87, lines 11-25)

A person having ordinary skill in the art would be able to derive computer instructions on a computer readable medium as recited in claim 9, as well as dependent claims 10-16, given **Figures 37** and **38,** and the corresponding description in the specification at page 80, line 19, to page 88, line 2, without undue experimentation.

Dependent claim 10:

The present invention provides fifth instructions for arithmetically combining values of the hardware counters to generate a combined counter value. (Specification, page 87, lines 14-15)

The present invention provides sixth instructions for comparing the combined counter value to the threshold value. (Specification, page 87, lines 15-16) The present invention provides seventh instructions for performing the action in response to a relationship between the combined counter value and the threshold value being present. (Specification, page 87, lines 20-23)

Dependent claim 11:

The present invention provides for the action including generating an interrupt if the predetermined relationship between the combined counter value and the threshold value is present. (Specification, page 87, lines 20-23)

Dependent claim 12:

The present invention provides for the fifth, sixth and seventh instructions are executed in response to incrementing a hardware counter. (Specification, page 87, lines 4-25; page 81, lines 4-13; and page 81, line 23 to page 82, line 19)

Dependent claim 13:

The present invention provides for the fifth, sixth and seventh instructions are executed within microcode of a processor of the data processing system. (Specification, page 87, lines 4-25; page 81, lines 4-13; and page 81, line 23 to page 82, line 19)

Dependent claim 14:

The present invention provides eighth instructions for sending the interrupt to an interrupt handler of a performance monitoring application, wherein the interrupt handler performs an operation based on receipt of the interrupt. (Specification, page 87, lines 23-24; page 81, lines 4-13; and page 84, lines 6-10)

Dependent claim 15:

The present invention provides for the operation is at least one of generating a log entry in a performance monitoring application log and notifying a log daemon process of an event. (Specification, page 84, lines 9-14)

Dependent claim 16:

The present invention provides for the fifth instructions for arithmetically combining values of the hardware counters includes instructions for combining values in accordance with a condition indicated by a performance monitoring application. (Specification, page 81, lines 4-13; and page 84, lines 19-24)

Independent claim 17:

The present invention provides for an apparatus for processing instructions. (Specification, page 81, lines 4-13) The present invention provides means for receiving a threshold value and an identification of a plurality of addresses to be monitored during the execution of a computer program. (Specification, page 86, line 27, to page 87, line 1) The present invention provides means for associating hardware counters with the plurality of addresses. (Specification, page 87, lines 4-6) The present invention provides means for executing the computer program and incrementing respective hardware counters when the plurality of addresses are accessed and a performance indicator associated with the plurality of addresses is encountered. (Specification, page 87, lines 4-6) The present invention provides means for performing an action in response to a determination that a predefined relationship between the threshold value and a combination of values obtained from the hardware counters is present. (Specification, page 87, lines 11-25)

The apparatus recited in claim 17, as well as dependent claims 17-23, may be an apparatus comprised of a performance monitoring unit 240 with hardware counters 241 and 242 and registers 243 and 244 of Figure 2. Hardware counters 241 and 242 and registers 243 and 244 of Figure 2 are the same as hardware counters 3720 and 3740 of Figure 37 and registers 243

and 244 of Figure 2 are the same as register 3750 of Figure 37. The counters, registers, and performance monitoring unit perform the steps described in the specification at page 80, line 19, to page 88, line 2, without undue experimentation.

Dependent claim 18:

The present invention provides means for arithmetically combining values of the hardware counters to generate a combined counter value. (Specification, page 87, lines 14-15) The present invention provides means for comparing the combined counter value to the threshold value. (Specification, page 87, lines 15-16) The present invention provides means for performing the action in response to a relationship between the combined counter value and the threshold value being present. (Specification, page 87, lines 20-23)

Dependent claim 19:

The present invention provides for the action including generating an interrupt if the predetermined relationship between the combined counter value and the threshold value is present. (Specification, page 87, lines 20-23)

Dependent claim 20:

The present invention provides for the means for arithmetically combining values of the hardware counters, means for comparing the combined couther value to the threshold value, and means for performing the action operate in response to incrementing a hardware counter. (Specification, page 87, lines 4-25; page 81, lines 4-13; and page 81, line 23 to page 82, line 19)

Dependent claim 21:

The present invention provides means for sending the interrupt to an interrupt handler of a performance monitoring application, wherein the interrupt handler performs an operation based on receipt of the interrupt. (Specification, page 87, lines 23-24; page 81, lines 4-13; and page 84, lines 6-10)

Dependent claim 22:

The present invention provides for the operation is at least one of generating a log entry in a performance monitoring application log and notifying a log daemon process of an event. (Specification, page 84, lines 9-14)

Dependent claim 23:

The present invention provides for the means for arithmetically combining values of the hardware counters includes means for combining values in accordance with a condition indicated by a performance monitoring application. (Specification, page 81, lines 4-13; and page 84, lines 19-24)

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The grounds of rejection to review on appeal are as follows:

A. GROUND OF REJECTION (Claims 1-2, 4, 8-10, 12, 16-18, 20, and 23)

Whether claims 1-2, 4, 8-10, 12, 16-18, 20, and 23 are obvious under 35 U.S.C. § 103(a) over Pekarich et al. (U.S. Patent No. 6,549,998 B1) in view of IBM Technical Disclosure "Hardware Cycle Based Memory Residency," hereinafter "IBMTD," and in further view of Burrows (U.S. Patent 5,887,159).

B. GROUND OF REJECTION (Claims 3, 6, 11, 14, 19, and 21)

Whether claims 3, 6, 11, 14, 19, and 21 are obvious under 35 U.S.C. § 103(a) over Pekarich et al. (U.S. Patent No. 6,549,998 B1) in view of IBM Technical Disclosure, "Hardware Cycle Based Memory Residency," hereinafter "IBMTD," and in further view of Burrows (U.S. Patent 5,887,159) as applied to claims 1, 9, and 17 above, and further in view of Levine et al. (U.S. Patent No. 5,797,019).

C. GROUND OF REJECTION (Claims 7, 15, and 22)

Whether claims 7, 15, and 22 are obvious under 35 U.S.C. § 103(a) over Pekarich et al. (U.S. Patent No. 6,549,998 B1), IBM Technical Disclosure, "Hardware Cycle Based Memory Residency," hereinafter "IBMTD," in further view of Burrows (U.S. Patent 5,887,159), and in view of Levine et al. (U.S. Patent No. 5,797,019) as applied to claims 3, 11, and 19 above, and further in view of Bartfai et al. (U.S. Patent Application Publication No. 2003/0101367 A1).

D. GROUND OF REJECTION (Claims 5 and 13)

Whether claims 5 and 13 are obvious under 35 U.S.C. § 103(a) over Pekarich et al. (U.S. Patent No. 6,549,998 B1) in view of IBM Technical Disclosure, "Hardware Cycle Based Memory Residency," hereinafter "IBMTD," in further view of Burrows (U.S. Patent 5,887,159) as applied to claims 1 and 9 above, and further in view of Randall Hyde, "The Art of Assembly Language."

ARGUMENT

A. 35 U.S.C. § 103, Obviousness, Claims 1-2, 4, 8-10, 12, 16-18, 20, and 23

A.1. Group A: Claims 1, 9, and 17

Claim 1 is representative of the claims in this group and reads as follows:

1. A method in a data processing system for processing instructions, the method comprising:

receiving a threshold value and an identification of a plurality of addresses to be monitored during the execution of a computer program;

associating hardware counters with the plurality of addresses;

executing the computer program and incrementing respective hardware counters when the plurality of addresses are accessed **and a performance indicator associated with the plurality of addresses is encountered**; and

performing an action in response to a determination that a predefined relationship between the threshold value and a combination of values obtained from the hardware counters is present. (emphasis added)

Pekarich, IBMTD and Burrows, taken alone or in combination, fail to teach or suggest executing the computer program and incrementing respective hardware counters when the plurality of addresses are accessed **and a performance indicator associated with the plurality of addresses is encountered**.

In the abstract, Pekarich states:

> An interleaver generates a valid interleaved data address for each iteration i of the mapping by the interleaver without employing a multiplication operation. The interleaver includes an address generator comprises two counters, bit-reverse and index tables, and an accumulation register array. The interleaver further comprises two adders, two registers storing tentative address values address.sub.i and address.sub.i+1, and select logic including a comparator, two buffers, and a multiplexer (mux). Two counters are employed to allow the interleaver to generate at least one valid address for each iteration, and a tentative address is generated from each output value of the two counters. Each iteration generates an output interleaved address. A tentative address is generated by using a portion of the counter value as an address to select a corresponding entry from each of the bit-reverse and index tables and the accumulation register array. The selected values from the index table and accumulation register array are combined in an adder. The value selected from the bit-reverse table is appended to the combination of the selected values from the index table and accumulation register

> array to form the tentative address. The tentative address generated from the first counter value is compared with a threshold value, and, based on the comparison, one of the two tentative addresses is selected as the output interleaved address. Before beginning the next iteration, the accumulated value used in generating the valid output interleaved address is updated to a new accumulated value. If not all output interleaved addresses have been generated, the counters are incremented by the same increment value, the increment value dependent upon the comparison with the threshold value, and the next iteration begins.

Thus, Pekarich is directed to an address generator for interleaving data. In Pekarich, an interleaver generates a valid interleaved data address for each iteration of the mapping by the interleaver without employing a multiplication operation. The interleaver generates at least one valid address for each iteration, and a tentative address is generated from each output value of the two counters. Each iteration generates an output interleaved address.

Thus, Pekarich merely describes an interleaver that generates an interleaved address for one sequence of data values by generating at least two counter values, each counter value having a predetermined offset from each other counter value. Thus, while Pekarich may teach a threshold and an address, Pekarich is not directed to processing instructions and does not teach or suggest executing the computer program and incrementing respective hardware counters when the plurality of addresses are accessed **and a performance indicator associated with the plurality of addresses is encountered**.

The Office Action acknowledges that Pekarich does not disclose executing the computer program and incrementing respective hardware counters when plurality of addresses are accessed and a performance indicator associated with the plurality of addresses is encountered; however, the Office Action alleges that IBMTD discloses associating hardware counters with the plurality of addresses. On page 1, paragraph 1, IBMTD states:

> The problem solved by this invention is the overhead of scanning memory and having pages stay resident in memory for long periods of times. One solution addresses this by releasing pages after they are read in or written out by the application, but this takes away the cache rehits, if they should occur. Another solution addresses memory-resident pages by running a daemon at a given interval, but this daemon, syncd, scans all of memory, and if there are a large number of pages to be written out to disk, it can cause considerable slowdown, in addition the period of time between scans may be too long if the cache rehit rate takes place in a relatively small amount of time. The current method, LRU, does a two pass scan. On the first pass, if the page has been recently referenced, it will

> reset the reference bit, and then move to the next page. If the next page is free it will give it to the requesting process. If there are no pages, it will do a second pass through memory.

Thus, IBMTD is directed to the overhead of scanning memory and having pages stay resident in memory for long periods of time. IBMTD introduces a method of freeing pages from memory while allowing for the probability of cache hits. Thus, IBMTD is directed to clearing pages from memory that has been resident to long periods.

IBMTD describes that, when a page is requested, a page frame table (PFT) for that page will be updated with the counter value for the hardware cycle counter. The Office Action alleges that "each page is associated with a page frame table which in turn is associated with physical addresses." Thus, using the Office Action's association, IBMTD allegedly teaches that each time an address is accessed a hardware counter value is updated. The present claim recites "executing the computer program and incrementing respective hardware counters when the plurality of addresses are accessed **and a performance indicator associated with the plurality of addresses is encountered.**"

Furthermore, IBMTD does not teach or suggest performing an action in response to a determination that a predefined relationship between the threshold value and a combination of values obtained from the hardware counters is present. As discussed above, the hardware counters of IBMTD are incremented each time an address is accessed, thus, the action performed by IBMTD is performed in response to a hardware counter that identifies address accesses and not a hardware counter that identifies when the plurality of addresses are accessed **and** a performance indicator associated with the plurality of addresses is encountered.

The Office Action acknowledges that Pekarich and IBMTD do not expressly disclose executing the computer program and incrementing respective hardware counters when plurality of addresses are accessed and a performance indicator associated with the plurality of addresses is encountered; however, the Office Action alleges that Burrows does. Burrows is directed to locating hint fields embedded within instructions and replacing the hint fields with calls to intercept the execution flow and redirect to procedures of a monitor. During code execution, Burrows records the existing hint information in a memory and analyzes the hint information to determine the most frequently occurring or best_hint value. When a best_hint value has been

determined, Burrows restores the replaced instructions with best_hint values. (see Burrows, Abstract)

In column 2, lines 49-65, Burrows states:

> FIG. 1 shows a process 100 which can be used for dynamically determining hint fields of instructions of machine executable code. A programmer generates, for example, object-oriented source programs 110 using conventional programming techniques. The source programs 110, once processed, are intended for execution in a computer system (CPU) 190. The programs 110 can be compiled by a compiler 111 into object code modules (obj) 120. The object code modules 120 include instructions with hint fields. Hint fields help branch prediction logic of the CPU 190 to determine the address of a next instruction to be fetched. Execution cycles are saved if the instructions are correctly fetched. Instructions whose destination addresses can only be resolved at run-time have their hint fields set to null.
>
> A monitor program 130 is also generated. The purpose of the monitor program 130 is to dynamically intercept the execution flow of the object modules 120 to record and analyze hint information.

In this section, Burrows describes determining which instructions have hint fields. The Office Action equates Burrows' hint field to the presently claimed indicator. Burrows' hint fields are described as fields that help branch prediction logic of the CPU to determine the address of a next instruction to be fetched. Thus, Burrows' hint field specifies a likely target address. In contradistinction, the present invention in claim 1 provides an indicator that identifies the instruction as one that is to be monitored by a performance monitor unit. Therefore, Burrows does not teach or suggest executing the computer program and incrementing respective hardware counters **when** the plurality of addresses are accessed **and a performance indicator associated with the plurality of addresses is encountered**, as recited in claim 1.

Additionally, the Office Action alleges that Burrows teaches executing the computer program and incrementing respective hardware counters when the plurality of addresses are accessed **and a performance indicator associated with the plurality of addresses is encountered** in the following section and figure:

> The hit field 520 can be incremented and decremented. The best_hint field 540 yields a value which can be used in the hint field 220 of the replaced instruction. The value 540 is dynamically determined as the machine code is executing.

(Burrows, column 5, lines 13-17)

In the case of a hit, a determination is made (645) if the current hit is the identical to the one stored in the best_hint field. If the current hit is the same, then increment the hit field (650).

(Burrows, column 5, lines 31-34)

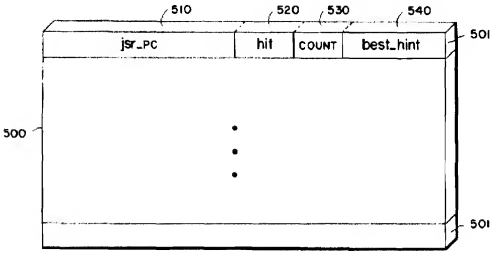

FIG. 5

(Burrows, Figure 5)

In these sections and figure, Burrows describes a counter for a hint field that can be incremented or decremented and a best_hint field that is a most frequently occurring hint field (see Burrows, Abstract). Burrows describes a hint field as fields that help branch prediction logic of the CPU to determine the address of a next instruction to be fetched. Thus, Burrows' hint field specifies a likely target address. Appellants respectfully submit that Burrows teaches that all of the hint fields associated with an address are counted in order to establish a best_hint field, so that the best_hint field may replace the hint field in the instruction. Thus, Burrows counts all of the hint fields and not just the first events. Consequently, Burrows teaches away from the present invention by counting every hint field and Burrows does not teach or suggest executing the computer program and incrementing respective hardware counters **when** the plurality of addresses are accessed **and a performance indicator associated with the plurality of addresses is encountered**. Thus, Appellants' hardware counters are updated when an address is accessed **and** a performance indicator associated with the address is encountered.

The Office Action bears the burden of establishing a *prima facie* case of obviousness based on the prior art when rejecting claims under 35 U.S.C. § 103. *In re Fritch*, 972 F.2d 1260, 23 U.S.P.Q.2d 1780 (Fed. Cir. 1992). Since the references fail to teach or suggest executing the computer program and incrementing respective hardware counters when the plurality of addresses are accessed **and a performance indicator associated with the plurality of addresses is encountered**, the Office Action has failed to establish a *prima facie* case of obviousness, because the Office Action does not show where each and every claim limitation is taught or fairly suggested by the applied prior art.

The applied references do not teach or suggest each and every claim limitation; therefore, Pekarich, IBMTD, and Burrows, taken alone or in combination, do not render claim 1 obvious. Independent claims 9 and 17 recite similar subject matter addressed above with respect to claim 1 and are allowable for similar reasons. Since claims 2-8, 10-16, and 18-23 depend from claims 1, 9, and 17, the same distinctions between Pekarich, IBMTD, and Burrows and the invention recited in claims 1, 9, and 17 apply for these claims. Additionally, claims 2-8, 10-16, and 18-23 recite other additional combinations of features not taught or suggested by the references.

Furthermore, no suggestion is present in any of the references to modify the references to include such features. That is, there is no teaching or suggestion in Pekarich, IBMTD, and Burrows that a problem exists for which executing the computer program and incrementing respective hardware counters when the plurality of addresses are accessed **and a performance indicator associated with the plurality of addresses is encountered**, is a solution. To the contrary, Pekarich appears to teach an interleaver that generates an interleaved address for one sequence of data values by generating at least two counter values, IBMTD appears to teach overhead of scanning memory and having pages stay resident in memory for long periods of time, and Burrows appears to teach counting every hint field.

Moreover, none of the reference teaches or suggests the desirability of incorporating the subject matter of the other reference. That is, there is no motivation offered in either reference for the alleged combination. The Office Action alleges that the motivation would be "to reduce the expense of operations by allowing more immediate and more cost-effective memory management with the use of a hardware cycle counter and PFT cycle counter" and "to measurably improve system performance by making code run faster." The present invention

provides for executing a computer program and incrementing respective hardware counters when the plurality of addresses are accessed and a performance indicator associated with the plurality of addresses is encountered. As discussed above, Pekarich appears to teach an interleaver that generates an interleaved address for one sequence of data values by generating at least two counter values, IBMTD appears to teach freeing pages from memory while allowing for the probability of cache hits, and Burrows appears to teach counting every hint field. None of the reference teaches or suggests executing the computer program and incrementing respective hardware counters when the plurality of addresses are accessed and a performance indicator associated with the plurality of addresses is encountered. Thus, the only teaching or suggestion to even attempt the alleged combination is based on a prior knowledge of Appellants' claimed invention thereby constituting impermissible hindsight reconstruction using Appellants' own disclosure as a guide.

One of ordinary skill in the art, being presented only with Pekarich, IBMTD, and Burrows, and without having a prior knowledge of Appellants' claimed invention, would not have found it obvious to combine and modify Pekarich, IBMTD, and Burrows to arrive at Appellants' claimed invention, as recited in claim 1. To the contrary, even if one were somehow motivated to combine Pekarich, IBMTD, and Burrows, and it were somehow possible to combine the systems, the result would not be the invention, as recited in claim 1. The resulting system would still fail to execute the computer program and increment respective hardware counters when the plurality of addresses are accessed **and** a performance indicator associated with the plurality of addresses is encountered.

In view of the above, Appellants respectfully submit that the Pekarich, IBMTD, and Burrows, taken alone or in combination, fail to teach or suggest the features of claims 1, 9, and 17. At least by virtue of their dependency on claims 1, 9, and 17, the features of dependent claims 2-8, 10-16, and 18-23 are not taught or suggested by Pekarich, IBMTD, and Burrows, whether taken individually or in combination. Accordingly, Appellants respectfully request the rejection of claims 1, 2, 4, 8-10, 12, 16-18, 20, and 23 under 35 U.S.C. § 103 not be sustained.

A.2. Group B: Claims 8, 16, and 23

With regard to claims 8, 16, and 23 in this group, Pekarich, IBMTD, and Burrows, taken alone or in combination, do not teach or suggest arithmetically combining values of the hardware counters includes combining values in accordance with a condition indicated by a performance monitoring application. The Office Action alleges that IBMTD teaches this feature but then states "IBMTD does not specifically disclose a condition indicated by a performance monitoring application, however, it is inherently required that the conditions associated with the process of subtraction be stored somewhere within the system in order for difference calculation between to the "hardware counter" and "PFT counter" to be possible." Appellants are claiming that the values of the hardware counters are arithmetically combined when a condition indicated by a performance monitoring application is met. Appellants are not claiming a data structure for storing a difference in calculation as alleged by the Office Action. Thus, Pekarich and IBMTD, alone or in combination do not teach or suggest arithmetically combining values of the hardware counters includes combining values in accordance with a condition indicated by a performance monitoring application.

Thus, in addition to being dependent on independent claims 1, 9, and 17, the specific features of dependent claims 8, 16, and 23 are also distinguishable over Pekarich, IBMTD, and Burrows, either alone or in combination, by virtue of the specific features recited in these claims. Accordingly, Appellants respectfully request the rejection of dependent claims 8, 16, and 23 under 35 U.S.C. § 103 not be sustained.

B. 35 U.S.C. § 103, Obviousness, Claims 3, 6, 11, 14, 19, and 21

Claims 3, 6, 11, 14, 19, and 21 are dependent on independent claims 1, 9, and 17 and, thus, these claims distinguish over Pekarich, IBMTD, and Burrows for at least the reasons noted above with regards to claims 1, 9, and 17. Moreover, Levine do not provide for the deficiencies of Pekarich, IBMTD, and Burrows and, thus, any alleged combination of Pekarich, IBMTD, Burrows, and Levine would not be sufficient to reject independent claims 1, 9, and 17 or claims 3, 6, 11, 14, 19, and 21 by virtue of their dependency. That is, Levine does not teach or suggest executing the computer program and incrementing respective hardware counters when the

plurality of addresses are accessed **and a performance indicator associated with the plurality of addresses is encountered**.

In view of the above, Pekarich, IBMTD, Burrows, and Levine, taken either alone or in combination, fail to teach or suggest the specific features recited in independent claims 1, 9, and 17, from which claims 3, 6, 11, 14, 19, and 21 depend. Accordingly, Appellants respectfully request the rejection of claims 3, 6, 11, 14, 19, and 21 under 35 U.S.C. § 103 not be sustained.

V. 35 U.S.C. § 103, Alleged Obviousness – Claims 7, 15, and 22

Claims 7, 15, and 22 are dependent on independent claims 1, 9, and 17 and, thus, these claims distinguish over Pekarich, IBMTD, Burrows, and Levine for at least the reasons noted above with regards to claims 1, 9, and 17. Moreover, Bartfai does not provide for the deficiencies of Pekarich, IBMTD, Burrows, and Levine and, thus, any alleged combination of Pekarich, IBMTD, Burrows, Levine, and Bartfai would not be sufficient to reject independent claims 1, 9, and 17 or claims 7, 15, and 22 by virtue of their dependency. That is, Bartfai does not teach or suggest executing the computer program and incrementing respective hardware counters when the plurality of addresses are accessed **and a performance indicator associated with the plurality of addresses is encountered**.

In view of the above, Pekarich, IBMTD, Burrows, Levine, and Bartfai, taken either alone or in combination, fail to teach or suggest the specific features recited in independent claims 1, 9, and 17, from which claims 7, 15, and 22 depend. Accordingly, Appellants respectfully request the rejection of claims 7, 15, and 22 under 35 U.S.C. § 103 not be sustained.

VI. 35 U.S.C. § 103, Alleged Obviousness – Claims 5 and 13

Claims 5 and 13 are dependent on independent claims 1 and 9 and, thus, these claims distinguish over Pekarich, IBMTD, and Burrows for at least the reasons noted above with regards to claims 1 and 9. Moreover, Hyde does not provide for the deficiencies of Pekarich, IBMTD and Burrows and, thus, any alleged combination of Pekarich, IBMTD, Burrows, and Hyde would not be sufficient to reject independent claims 1 and 9 or claims 5 and 13 by virtue of their

dependency. That is, Hyde does not teach or suggest executing the computer program and incrementing respective hardware counters when the plurality of addresses are accessed **and a performance indicator associated with the plurality of addresses is encountered**.

In view of the above, Pekarich, IBMTD, Burrows, and Hyde, taken either alone or in combination, fail to teach or suggest the specific features recited in independent claims 1 and 9, from which claims 5 and 13 depend. Accordingly, Appellants respectfully request the rejection of claims 5 and 13 under 35 U.S.C. § 103 not be sustained.

CONCLUSION

In view of the above, Appellants respectfully submit that claims 1-23 are allowable over the cited prior art and that the application is in condition for allowance. Accordingly, Appellants respectfully request the Board of Patent Appeals and Interferences to reverse the rejections set forth in the Final Office Action.

Respectfully submitted,

/Francis Lammes/
Francis Lammes
Reg. No. 55,353
YEE & ASSOCIATES, P.C.
PO Box 802333
Dallas, TX 75380
(972) 385-8777

CLAIMS APPENDIX

The text of the claims involved in the appeal are:

1. A method in a data processing system for processing instructions, the method comprising:

receiving a threshold value and an identification of a plurality of addresses to be monitored during the execution of a computer program;

associating hardware counters with the plurality of addresses;

executing the computer program and incrementing respective hardware counters when the plurality of addresses are accessed and a performance indicator associated with the plurality of addresses is encountered; and

performing an action in response to a determination that a predefined relationship between the threshold value and a combination of values obtained from the hardware counters is present.

2. The method of claim 1, further comprising:

arithmetically combining values of the hardware counters to generate a combined counter value;

comparing the combined counter value to the threshold value; and

performing the action in response to a relationship between the combined counter value and the threshold value being present.

3. The method of claim 2, wherein the action includes:

generating an interrupt if the predetermined relationship between the combined counter value and the threshold value is present.

4. The method of claim 2, wherein the steps of arithmetically combining values of the hardware counters, comparing the combined counter value, and performing the action are performed in response to incrementing a hardware counter.

5. The method of claim 2, wherein the steps of arithmetically combining values of the hardware counters, comparing the combined counter value, and performing the action are performed within microcode of a processor of the data processing system.

6. The method of claim 3, further comprising sending the interrupt to an interrupt handler of a performance monitoring application, wherein the interrupt handler performs an operation based on receipt of the interrupt.

7. The method of claim 6, wherein the operation is at least one of generating a log entry in a performance monitoring application log and notifying a log daemon process of an event.

8. The method of claim 2, wherein arithmetically combining values of the hardware counters includes combining values in accordance with a condition indicated by a performance monitoring application.

9. A computer program product in a computer readable recordable-type medium for processing instructions comprising:

first instructions for receiving a threshold value and an identification of a plurality of addresses to be monitored during the execution of a computer program;

second instructions for associating hardware counters with the plurality of addresses;

third instructions for executing the computer program and incrementing respective hardware counters when the plurality of addresses are accessed and a performance indicator associated with the plurality of addresses is encountered; and

fourth instructions for performing an action in response to a determination that a predefined relationship between the threshold value and a combination of values obtained from the hardware counters is present.

10. The computer program product of claim 9, further comprising:

fifth instructions for arithmetically combining values of the hardware counters to generate a combined counter value;

sixth instructions for comparing the combined counter value to the threshold value; and

seventh instructions for performing the action in response to a relationship between the combined counter value and the threshold value being present.

11. The computer program product of claim 10, wherein the action includes:

generating an interrupt if the predetermined relationship between the combined counter value and the threshold value is present.

12. The computer program product of claim 10, wherein the fifth, sixth and seventh instructions are executed in response to incrementing a hardware counter.

13. The computer program product of claim 10, wherein the fifth, sixth and seventh instructions are executed within microcode of a processor of the data processing system.

14. The computer program product of claim 11, further comprising eighth instructions for sending the interrupt to an interrupt handler of a performance monitoring application, wherein the interrupt handler performs an operation based on receipt of the interrupt.

15. The computer program product of claim 14, wherein the operation is at least one of generating a log entry in a performance monitoring application log and notifying a log daemon process of an event.

16. The computer program product of claim 10, wherein the fifth instructions for arithmetically combining values of the hardware counters includes instructions for combining values in accordance with a condition indicated by a performance monitoring application.

17. An apparatus for processing instructions comprising:

means for receiving a threshold value and an identification of a plurality of addresses to be monitored during the execution of a computer program;

means for associating hardware counters with the plurality of addresses;

means for executing the computer program and incrementing respective hardware counters when the plurality of addresses are accessed and a performance indicator associated with the plurality of addresses is encountered; and

means for performing an action in response to a determination that a predefined relationship between the threshold value and a combination of values obtained from the hardware counters is present.

18. The apparatus of claim 17, further comprising:

means for arithmetically combining values of the hardware counters to generate a combined counter value;

means for comparing the combined counter value to the threshold value; and

means for performing the action in response to a relationship between the combined counter value and the threshold value being present.

19. The apparatus of claim 18, wherein the action includes:

generating an interrupt if the predetermined relationship between the combined counter value and the threshold value is present.

20. The apparatus of claim 18, wherein the means for arithmetically combining values of the hardware counters, means for comparing the combined couther value to the threshold value, and means for performing the action operate in response to incrementing a hardware counter.

21. The apparatus of claim 19, further comprising means for sending the interrupt to an interrupt handler of a performance monitoring application, wherein the interrupt handler performs an operation based on receipt of the interrupt.

22. The apparatus of claim 21, wherein the operation is at least one of generating a log entry in a performance monitoring application log and notifying a log daemon process of an event.

23. The apparatus of claim 18, wherein the means for arithmetically combining values of the hardware counters includes means for combining values in accordance with a condition indicated by a performance monitoring application.

EVIDENCE APPENDIX

There is no evidence to be presented.

RELATED PROCEEDINGS APPENDIX

There are no related proceedings.